

Modélisation géométrique et Synthèse d'images

Introduction

Corentin SOUTON



- Perception lumière / couleur
- Systèmes d'acquisition
- Imagerie numérique
- Représentation 3D
- Exemple d'applications
- Exemple de logiciels
- Conclusion



- **Images : définition**

Représentation d'une ou plusieurs êtres / objets par le dessin, la peinture, la sculpture, la gravure, la photographie, la cinématographie, etc.

Physique : représentation d'un objet, produite par la réunion des rayons ou faisceaux lumineux qui en émanent et se reconstituent sur un miroir, sur un écran, ou sur l'œil qui perçoit cette image.

- **Vision : définition**

Perception du monde extérieur par les organes de la vue.

Subjectif : chaque individu voit de manière différente la même scène

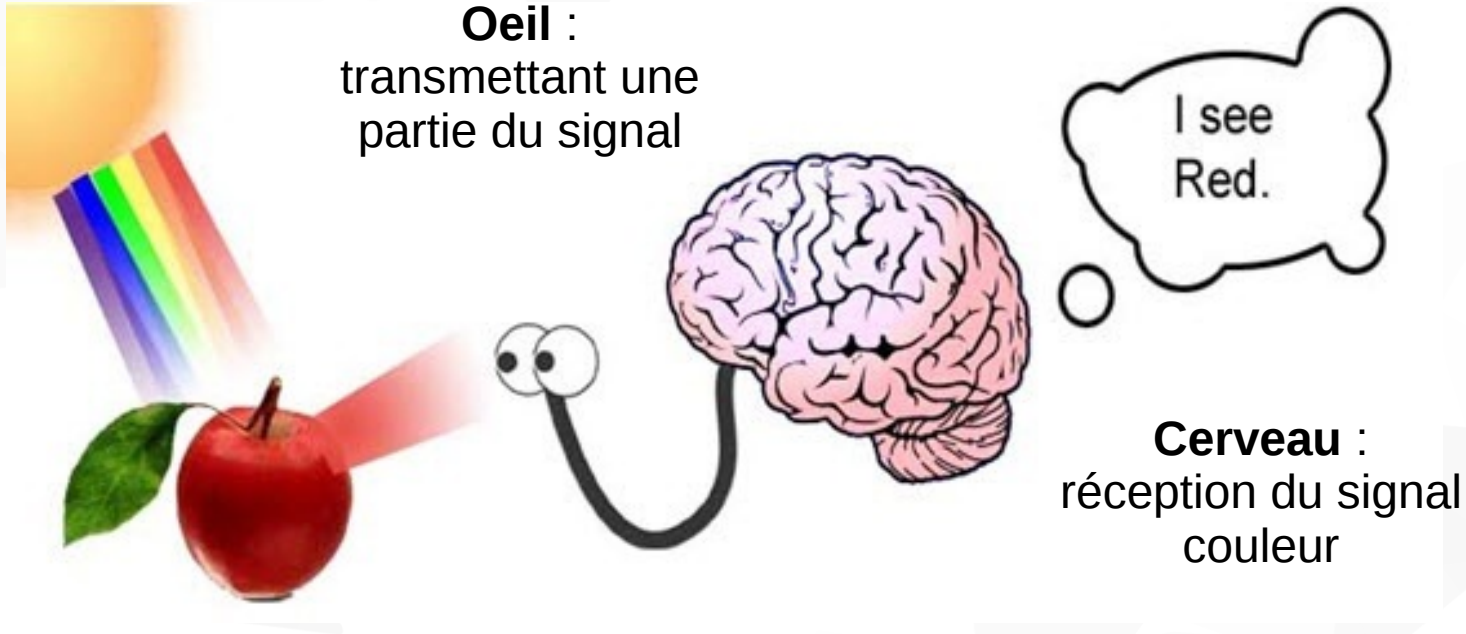
Machine de vision : forcément objectif devant une image.





Perception lumière / couleur

Source lumineuse :
rayonnement
électromagnétique



Objet : réfléchit les
rayons dans
plusieurs directions





Systemes d'acquisition

- **Éléments d'optique géométrique**

Rôle : projeter une scène 3D sur un plan 2D

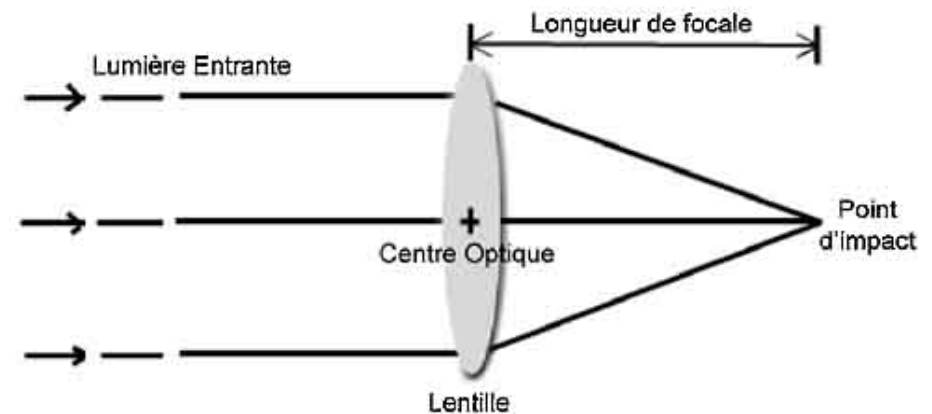
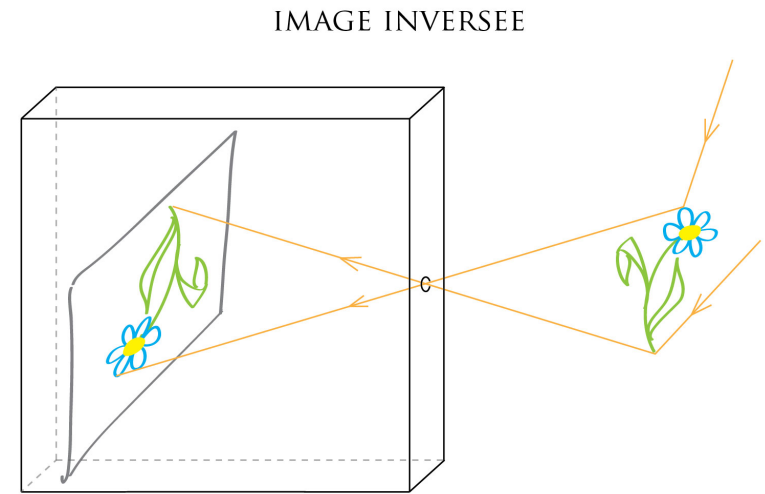
- **Sténopé**

15ème siècle

Inconvénients : effet de flou et diffraction

- **Lentille**

Inconvénients : seuls les points à une distance donnée de la lentille sont au focus





Imagerie numérique

- **Types d'images**

Images d'émission : éléments de la scène émettent les rayons perçus par le capteur (pas de source lumineuse)



Images de réflexion : rayons de la source sont réfléchis (ou diffusés) par un objet, puis atteignent le capteur



Images d'absorption : on s'intéresse aux rayons absorbés ou transmis





Imagerie numérique

- **Domaines d'application**

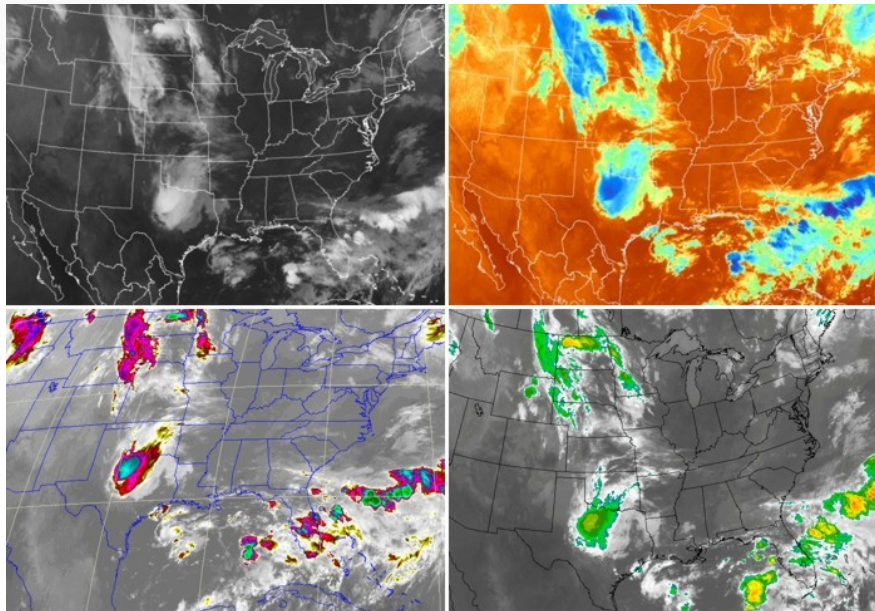
Infographie :

méthodes de création d'images numérique



Traitement d'image :

analyse d'image afin d'en extraire des informations





Représentation 3D

- **Lithophanie**

Surface rugueuse

Impression 3D



- **Stéréoscopie**

Superposition d'images

Lunettes ou
un projecteur numérique



ORIGINAL

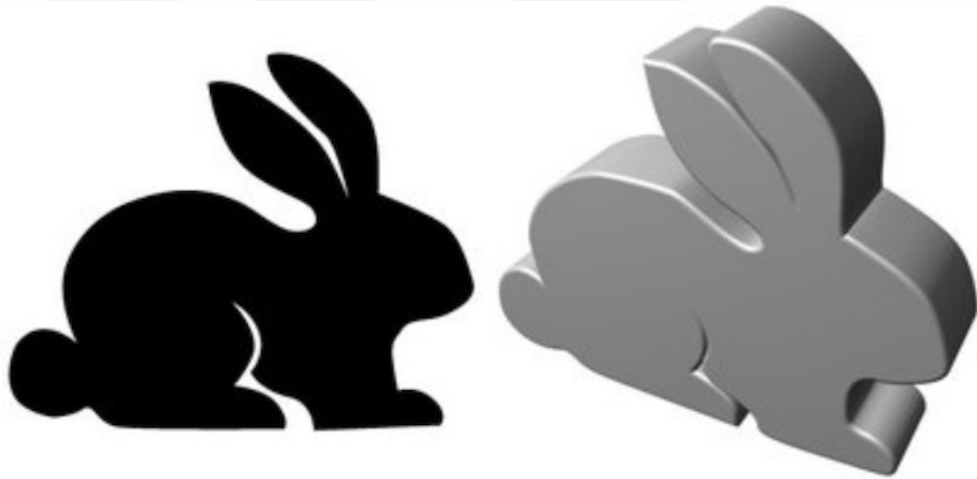


Stereoscopic 3D image





Représentation 3D



- **Synthèse d'image**

Application de formule géométrique

Layout (la scène)

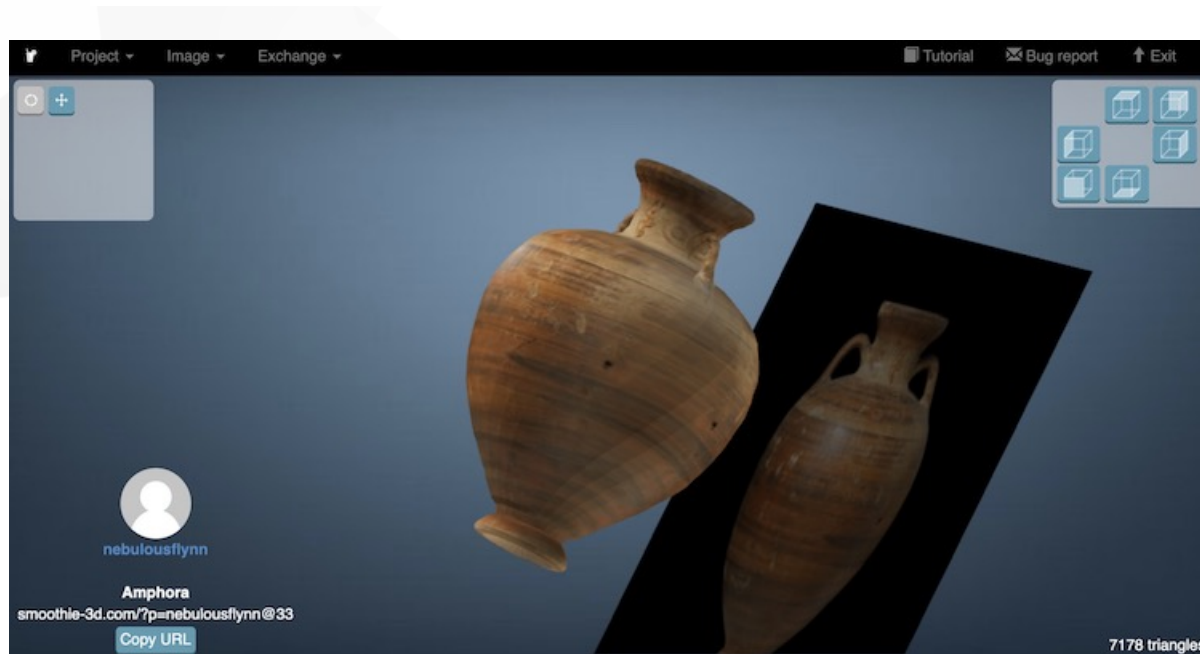
Rigging (squelettage)

Moteur physique

Position de la lumière

Shader (matériaux)

Rendu (calcul des images)





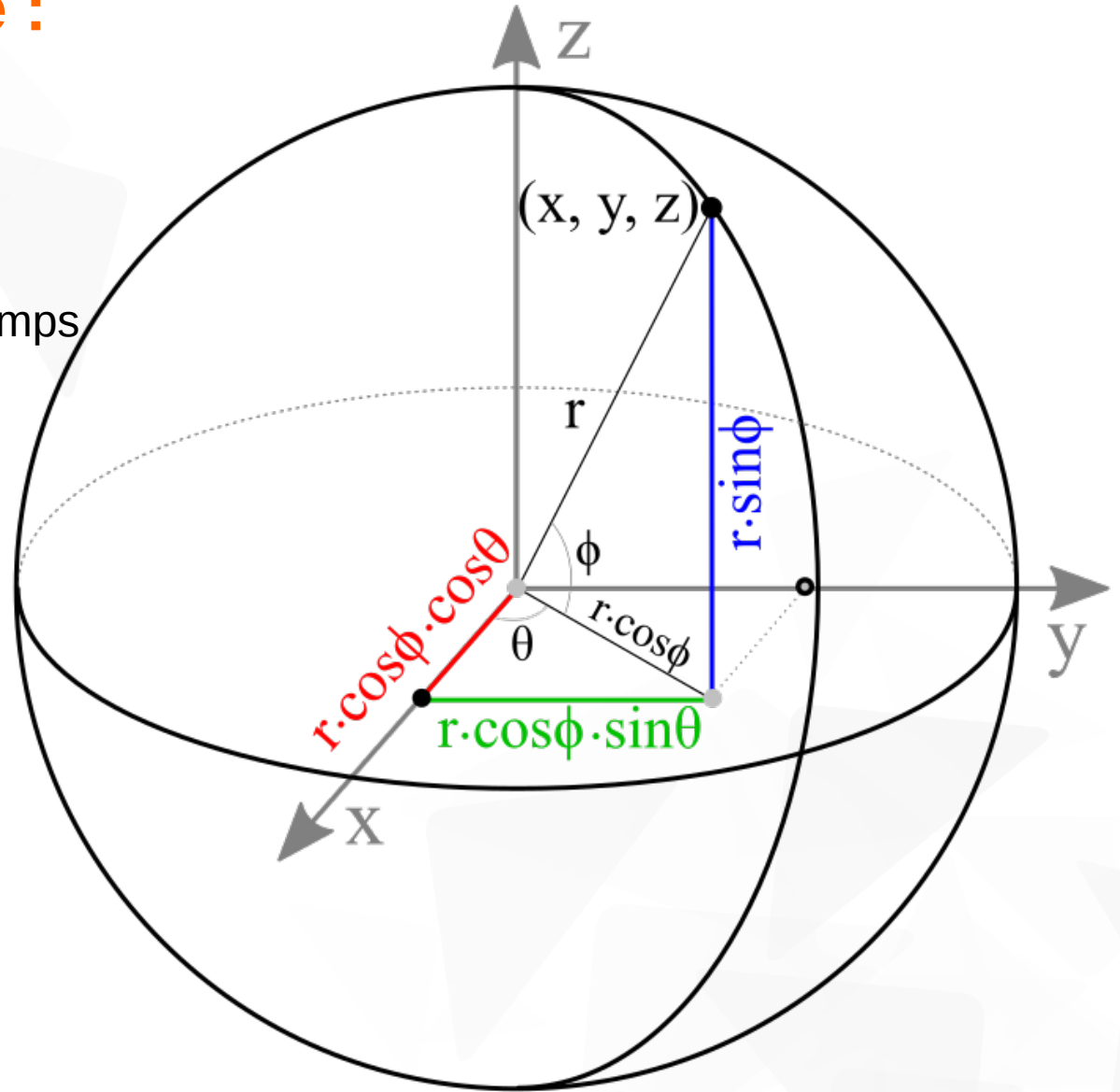
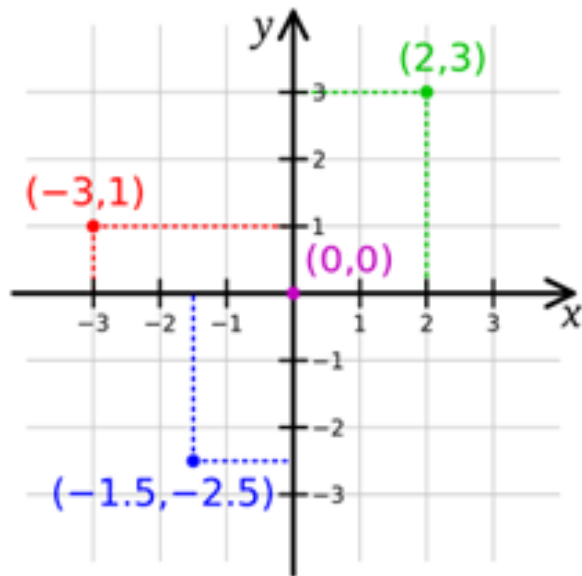
Représentation 3D

- **Systeme de coordonnée :**

Axes mutuellement perpendiculaire

Règles « de la main droite »

Cinématique dans l'espace : ajout du temps





Représentation 3D

- **Point**

$P = (x, y, z)$, où x , y et z sont des nombres

$O = (0,0,0)$, point origine

- **Représentation**

$P = O + aX + bY + cZ$

Exemple :

$$\begin{aligned}(2,3,4) &= (2,0,0) + (0,3,0) + (0,0,4) \\ &= (0,0,0) + (2,0,0) + (0,3,0) + (0,0,4) \\ &= (0,0,0) + 2(1,0,0) + 3(0,1,0) + 4(0,0,1) \\ &= O + 2X + 3Y + 4Z\end{aligned}$$

- **Vecteur**

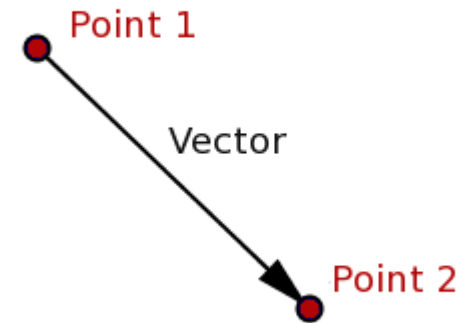
Espace entre deux points

3 vecteurs de bases :

$$X = (1,0,0)$$

$$Y = (0,1,0)$$

$$Z = (0,0,1)$$





Représentation 3D

Point Class

```
{  
Variables:  
num tuple[3]; //(x,y,z)
```

Operators:

```
Point AddVectorToPoint(Vector);  
Point SubtractVectorFromPoint(Vector);  
Vector SubtractPointFromPoint(Point);
```

Function:

```
//later this will call a function from a graphics API, but for now  
//this should just be printing the points coordinates to the screen
```

```
drawPoint;  
}
```





Représentation 3D

Vector Class

```
{  
Variables:  
num tuple[3]; //(x,y,z)
```

Operators:

```
Vector AddVectorToVector(Vector);  
Vector SubtractVectorFromVector(Vector);  
}
```

main

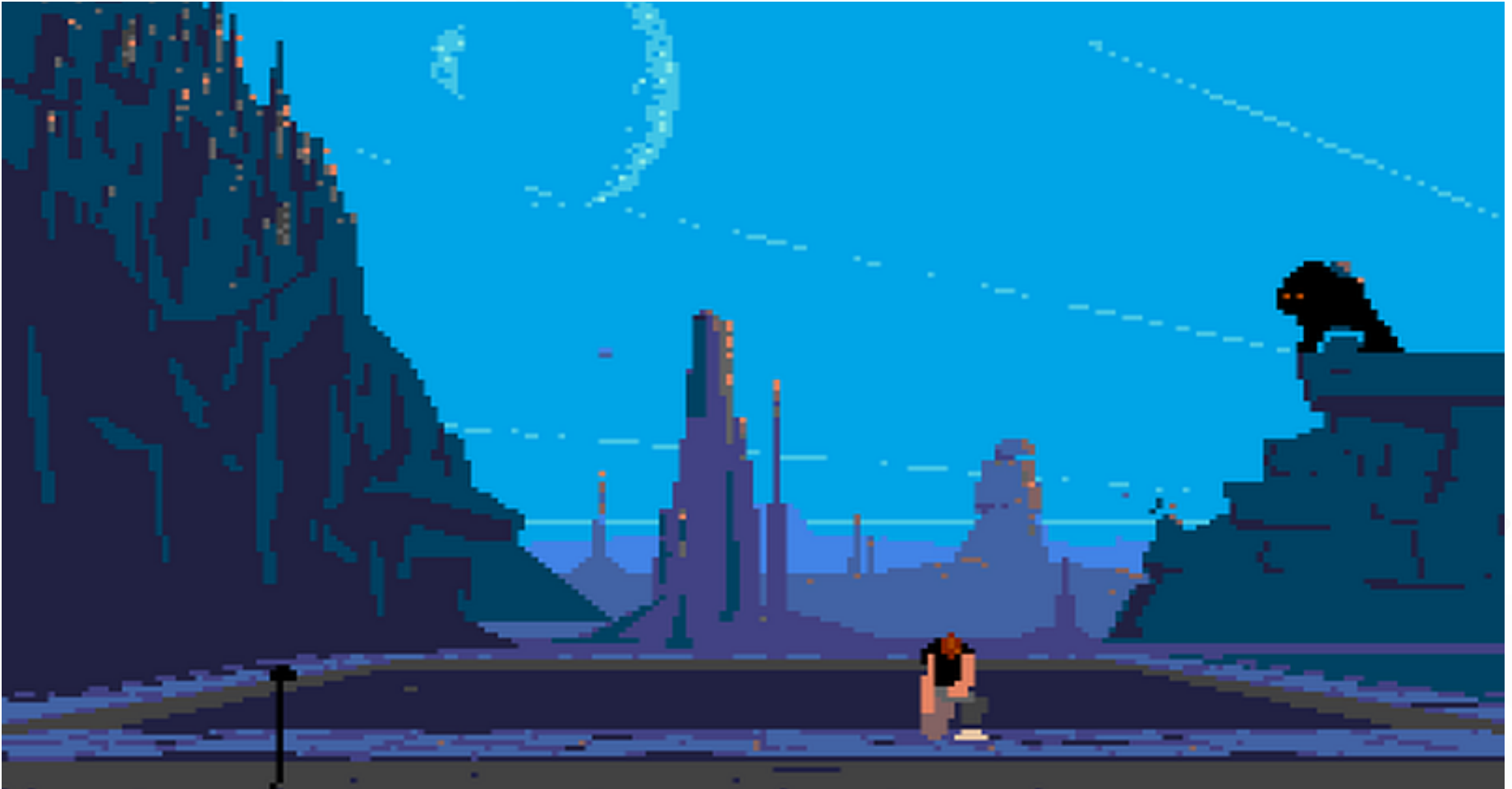
```
{  
    var point1 = new Point(1,2,1);  
    var point2 = new Point(0,4,4);  
    var vector1 = new Vector(2,0,0);  
  
    point1.drawPoint(); //should display (1,2,1)  
    point2.drawPoint(); //should display (0,4,4)  
}
```





Exemple d'application

- **Another World (1991)**





Exemple d'application

- Wolfenstein 3D (1992)





Exemple d'application

- Donkey Kong Country (1994)





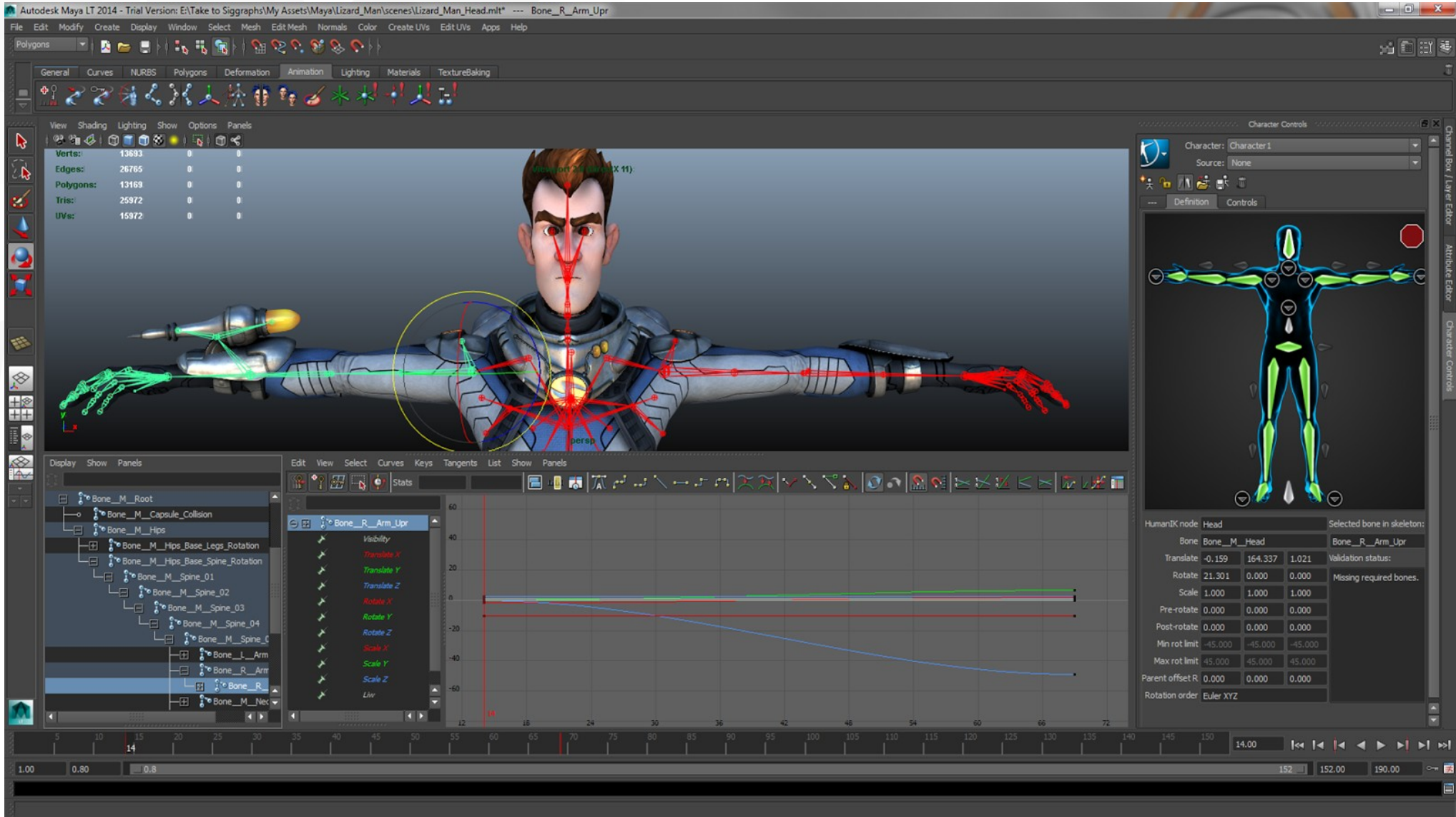
Exemple d'application

- **Quake II (1997)**



Exemple de logiciel

● PowerAnimator (Alias) et Maya

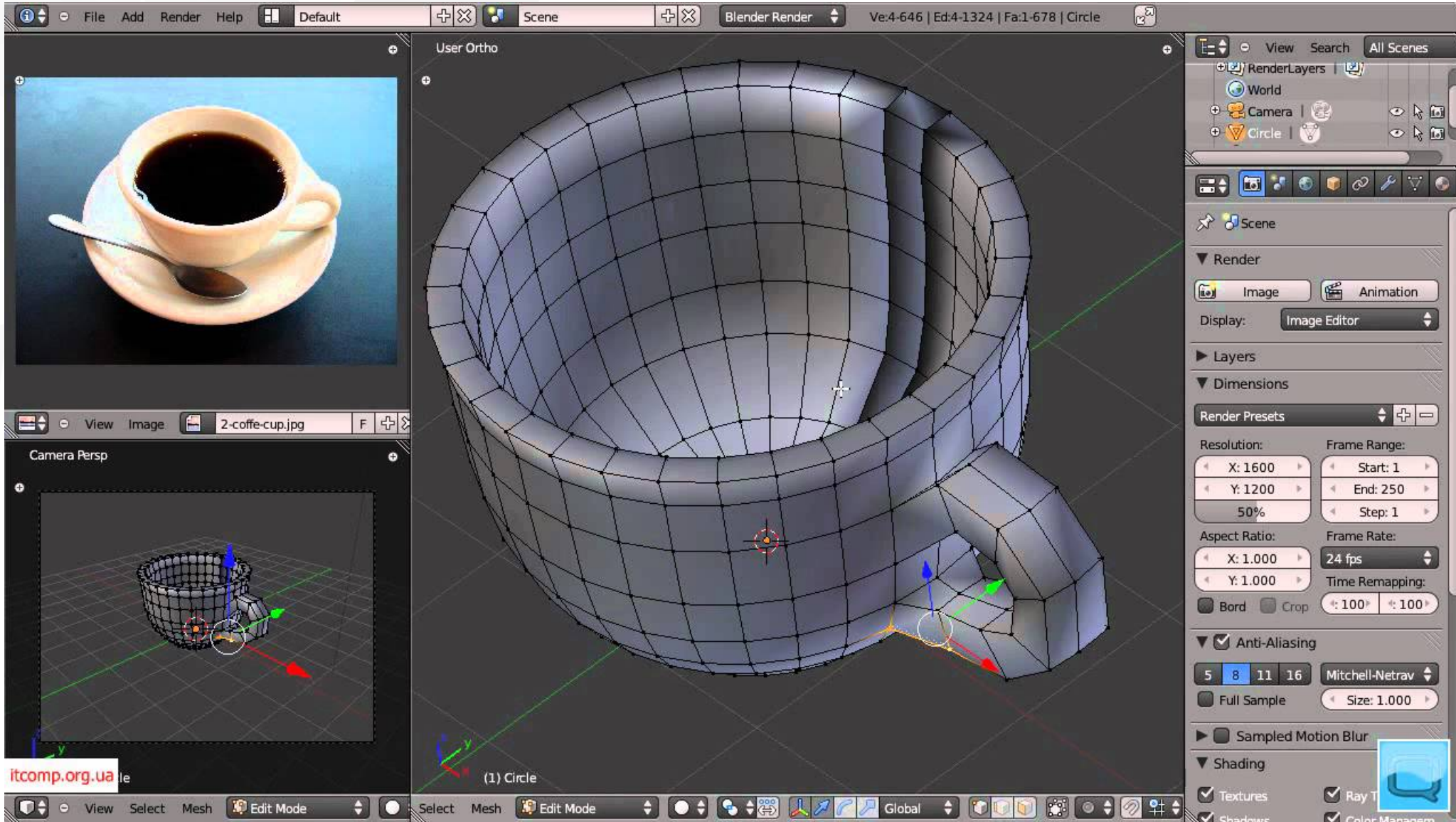




Exemple de logiciel

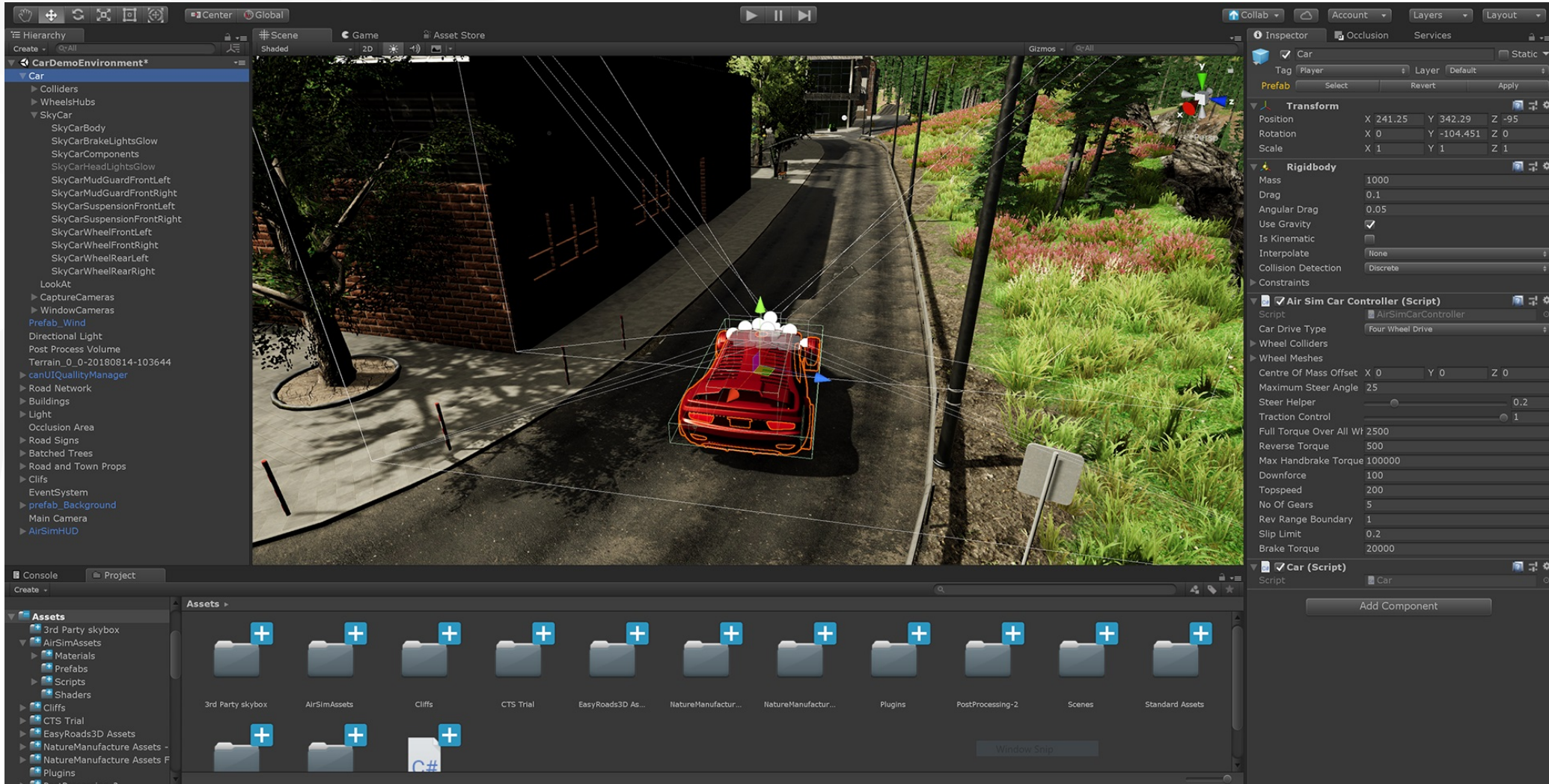


- **Blender 3D**








Exemple de logiciel

● Unity





Exemple de logiciel

Moteur	Langage	Propriétaire	Compagnie	Activité
	C, C++	Shareware Licence commercial	Autodesk	3D, films, animations
	C, C++, Python	Licence GPL	Blender Foundation	3D, films, animations, jeux vidéo
	C++, C#	Versions gratuites Licence commercial	Unity Technologies	2D, 3D jeux vidéo
	C++	Gratuit Open source	Epic Games	3D
	Javascript	Licence MIT	OpenFL Contributors	2D, jeux vidéo



FIN !

Merci ...

- ... pour avoir tenu jusqu'ici,
- ... de ne pas hésiter à poser des questions !

